Atty Docket No. 042390.P10677

United States Patent Application
in the Name of

Mark Schmisseur

for

SYSTEM AND METHOD FOR DEFINING PRIVATE FUNCTIONS OF A MULTI-FUNCTION PERIPHERAL DEVICE

Submitted by
BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
12400 Wilshire Blvd., Seventh Floor
Los Angeles, CA 90025-1026
Docket No. 042390.P10677

BACKGROUND

1. Field:

[0001] The subject matter disclosed herein relates to processing platforms. In particular aspects, the subject matter disclosed herein relates to processing platforms which transmit data in data busses.

Information:

[0002] A processing platform typically comprises a host processing system coupled to one or more peripheral devices through a data bus. Such a data bus may conform to any one of several industry standard architectures such as the peripheral component interconnect (PCI) standard. A peripheral device in a processing platform may be coupled to the device and define one or more device functions which are accessible through the data bus.

[0003] A processing platform may also comprise one or more peripheral devices to perform input/output (I/O) functions for a host processing system. Such peripheral devices may comprise an I/O controller and I/O devices defining one or more I/O channels associated with I/O formats such as, for example, redundant array of independent disks (RAID), Ethernet, Fibre-Channel, SSA and IBA. To support multiple I/O channels, a single I/O device may define multiple device functions which are accessible from a data bus. There is a need to determine systems and methods for controlling access to the multiple device functions in a peripheral device from a data bus.

BRIEF DESCRIPTION OF THE FIGURES

[0004] Non-limiting and non-exhaustive embodiments of the present invention will be described with reference to the following figures, wherein like reference numerals refer to like parts throughout the various figures unless otherwise specified.

[0005] Figure 1 shows a schematic of a processing platform according to an embodiment of the present invention.

[0006] Figure 2 shows a flow diagram of a process of establishing communication with two or more device functions of a peripheral device according to an embodiment of the present invention.

[0007] Figure 3 shows a schematic of a processing platform according to an alternative embodiment of the present invention.

[0008] DETAILED DESCRIPTION

[0009] Reference throughout this specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the present invention. Thus, the appearances of the phrase "in one embodiment" or "an embodiment" in various places throughout this specification are not necessarily all referring to the same embodiment. Furthermore, the particular features, structures, or characteristics may be combined in one or more embodiments.

[0010] "Machine-readable" instructions as referred to herein relates to expressions which may be understood by one or more machines for performing one or more logical operations. For example, machine-readable instructions may comprise instructions which are interpretable by a processor compiler for executing one or more operations one or more data objects. However, this is merely an example of machine-readable instructions and embodiments of the present invention are not limited in this respect.

[0011] "Storage medium" as referred to herein relates to media capable of maintaining expressions which are perceivable by one or more machines. For example, a storage medium may comprise one or more storage devices for storing machine-readable instructions. However, this is merely an example of a storage medium and embodiments of the present invention are not limited in this respect.

[0012] "Logic" as referred to herein relates to structure for performing one or more logical operations. For example, logic may comprise circuitry which provides one or more output signals based upon one or more input signals. Such circuitry may comprise a finite state machine which receives a digital input and provides a digital output, or circuitry which provides one or more analog output signals in response to one or more analog input signals. Also, logic may comprise processing circuitry in combination with machine-executable instructions stored in a storage medium. However, these are merely examples of structures which may provide logic and embodiments of the present invention are not limited in this respect.

[0013] A "processing system" as discussed herein relates to a combination of hardware and software resources for accomplishing computational tasks. However, this is merely an example of a processing system and embodiments of the present invention are not limited in this respect. A "host processing system" relates to a processing system which may be adapted to communicate with a "peripheral device." For example, a

peripheral device may provide inputs to or receive outputs from an application process hosted on the host processing system. However, these are merely examples of a host processing system and peripheral device, and embodiments of the present invention are not limited in these respects.

[0014] A "data bus" as referred to herein relates to circuitry for transmitting data between devices. For example, a data bus may transmit data between a host processing system and a peripheral device. However, this is merely an example of a data bus and embodiments of the present invention are not limited in this respect. A "bus transaction" as referred to herein relates to an interaction between devices coupled in a data bus structure wherein one device transmits data addressed to the other device through the data bus structure. However, this is merely an example of a bus transaction and embodiments of the present invention are not limited in this respect.

[0015] A "bridge" as referred to herein relates to a device coupled between data busses to transmit data between devices coupled to one bus and another bus. According to an embodiment, a bridge may be coupled between two busses for transmitting data between peripheral devices and processing resources. However, embodiments of the present invention are not limited in this respect and other applications of a bridge may be used. Also, a bridge may define a "primary" data bus which couples the bridge to a host processing system and define a "secondary" data bus which is opposite the host processing system. Such a bridge as described herein may be formed according to a peripheral components interconnection (PCI) as described in the PCI-to-PCI Bridge Architecture Specification, Rev. 1.1, December 18, 1998 (hereinafter "PCI-to-PCI Bridge Specification"). However, embodiments of the present invention are not limited in this respect and a bus, bridge or bus configuration according to other embodiments may be employed using other techniques.

[0016] A data bus may be coupled to one or more devices at "data interfaces" associated with addresses on the data bus. Such a data interface may comprise a physical connection to couple a device to the data bus. Also, a data interface may define a physical signaling format and an address to facilitate communication with an associated device in a bus transaction. However, these are merely examples of a data interface between a data bus and a device, and embodiments of the present invention are not limited in these respects.

[0017] A "device function" as referred to herein relates to an entity associated with a device coupled to a data bus at a data interface. The data bus may communicate with the device function through messages transmitted through the data interface. Also, multiple device functions may be associated with a single device such that a data bus may communicate with any particular device function through the transmission of signals and data through the data interface between the device and the data bus and addressed to the particular device function. However, these are merely examples of a device function and embodiments of the present invention are not limited in these respects.

[0018] "Bus enumeration" as referred to herein relates to a process including the identification of devices coupled to a data bus and allocating resources to communicate with identified devices. For example, a processing system coupled to a data bus may execute a bus enumeration process to identify devices on the data bus and any device functions provided by the identified devices, and allocate processing resources to communicate with the identified devices and device functions. However, this is merely an example of a bus enumeration process an embodiments of the present invention are not limited in this respect.

[0019] An "I/O channel" as referred to herein relates to an entity through which data may be transmitted to, or received from an external system. For example, an I/O channel may comprise a peripheral device or device function to transmit data between a data bus and a communication or storage device. However, this is merely an example of an I/O channel and embodiments of the present invention are not limited in this respect.

[0020] Briefly, an embodiment of the present invention relates to a system and method of enabling communication between device functions of a peripheral device and multiple processing systems. A peripheral device may define a plurality of device functions accessible through a data interface with a data bus. A first processing system may be adapted to communicate with a first device function of the peripheral device through the data interface with the peripheral device. A second processing system may be adapted to communicate with a second device function of the peripheral device through the data interface.

[0021] Figure 1 shows a schematic of a processing platform 10 according to an embodiment of the present invention. An I/O processor 14 provides a first processing

system for communicating with a peripheral device 16 and a host processing system 12 provides a second processing system for communicating with the peripheral device 16. The I/O processor 14 comprises an internal bridge which forms a primary bus 24 and a secondary bus 18. According to an embodiment, the primary and secondary busses 24 and 18 may be formed according to a PCI data bus structure such as that described in the PCI Local Bus Specification, Rev. 2.2, Dec. 18, 1998 published by the PCI Special Interest Group (hereinafter the "PCI Local Bus Specification"). However, this is merely an example of a bus structure which may be employed in a data bus to transmit data between devices and embodiments of the present invention are not limited in this respect. Also, the internal bridge may be formed according to the PCI-to-PCI Bridge Specification. However, this is merely an example of how a bridge may be implemented to form primary and secondary data busses in a processing platform and embodiments of the present invention are not limited in this respect.

[0022] The peripheral device 16 comprises a data interface with the secondary bus 18 to transfer data between processes at the peripheral device 16 and devices coupled to the secondary bus 18. Such a data interface may comprise any one of several data interfaces with a data bus such as, for example, a device "slot" on a PCI bus defined by a bus and device number as described in the PCI-to-PCI Bridge Specification at Chapter 13. Such a device slot may be associated with a signal definition and device pinout as described in chapter 2 and section 4.2.6 of the PCI Local Bus Specification. However, these are merely examples of how a peripheral device may comprise a data interface with a data bus and embodiments of the present invention are not limited in these respects.

[0023] In the illustrated embodiment, the peripheral device 16 may comprise an interface according to variations of the Small Computer System Interface (SCSI) established by the National Committee for Information Technology Standards (NCITS) to enable communication through first and second I/O channels 20 and 22. However, this is merely an example of how a peripheral device may facilitate communication with multiple I/O channels and other interfaces according to different formats such as, for example, Fibre-Channel, SSA, IBA or Ethernet. Each of the I/O channels 20 and 22 may be adapted to communicate with any one of several I/O devices such as, for example, a storage system such as a Redundant Array of Independent Disks (RAID), a communication port, a server, a client or other storage system directly or via a switch.

The peripheral device 16 comprises at least two device functions which may be adapted to communicate through respective I/O channels 20 and 22. However, this is merely an example of how a peripheral device may implement multiple device functions to provide multiple I/O channels and embodiments of the present invention are not limited in this respect.

[0024] In the illustrated embodiment, the I/O processor 14 and host processing system 12 may each execute an enumeration procedure to configure resources to communicate with one or more of the devices functions associated with the I/O channels 20 and 22. The I/O processor 14 may execute a first enumeration procedure to configure resources to communicate with a first device function and initiate a subsequent bus transaction to conceal one or more device functions from the host processing system 12. The host processing system 12 may then execute a subsequent enumeration procedure to configure resources to communicate with the second device function while not detecting the existence of the concealed device function(s).

[0025] According to an embodiment, the host processing system 12 and I/O processor 14 each comprise logic to execute a bus enumeration procedure to allocate resources to facilitate communication with one or more device coupled to the bus 24 or 18. For example, the host processing system 12 or I/O processor may comprise a processing system to initiate an enumeration procedure for configuring resources (e.g., allocation of local memory to data buffers for data transmitted between processes hosted on a processing system and a device or device function) by executing machine-readable instructions as a data bus driver stored in a storage medium. However, this is merely an example of how logic at the host processing system 12 or I/O processor 14 may execute an enumeration procedure and embodiments of the present invention are not limited in this respect.

[0026] Figure 2 shows a flow diagram of a process of enumerating two device functions of the peripheral device 16 according to an embodiment of the present invention. The I/O processor 14 may initiate a first bus transaction on the secondary bus 18 to configure resources to communicate with the first device function of the peripheral device 16 at block 102 and initiate a second bus transaction on the bus 18 to configure resources to communicate with the second device function. In an embodiment in which the secondary bus 18 comprises a PCI bus, blocks 102 and 104 may comprise initiating Type 0 configuration transactions to configure resources at the

I/O processor 14 to communicate with the first and second device functions as described in section 3.2.2.3.1 of the PCI Local Bus Specification. In an embodiment in which the primary bus 24 comprises a PCI bus, block 108 may comprise initiating a Type 1 configuration transaction on the primary bus 24 from the host processing system 12 (which may result in a Type 0 configuration transaction on the secondary bus 18) to configure resources at the host processing system 12 to communicate with one of the device functions associated with the peripheral device 16. However, this is merely an example of how a processing system may initiate a bus transaction to configure resources to communicate with a device function and embodiments of the present invention are not limited in this respect.

[0027] In the illustrated embodiment, the I/O processor 14 initiates a subsequent bus transaction at block 106 to conceal the first device function from the host processing system 12 when the host processing system 12 initiates a bus transaction to configure resources to communicate with the peripheral device 16 at block 108. Thus, the host processing system 12 may only configure resources to communicate with the unconcealed device function(s). In the illustrated embodiment, the I/O processor 14 initiates bus transactions to configure resources to communicate with both device functions. In alternative embodiments, the I/O processor 14 may only configure resources to communicate with the device function that is concealed from the host processing system 12 at block 106 (e.g., omitting the bus transaction at block 104) such that the I/O processor 14 and host processor 12 each exclusively communicate with one of the device functions of the peripheral device 16. However, these are merely examples of how device functions may be allocated to multiple processing system coupled to a data bus and embodiments of the present invention are not limited in these respects.

[0028] In an embodiment in which the secondary bus 18 comprises a PCI bus, a Type 0 configuration transaction may obtain information identifying the peripheral device 16 as a multi-function device from a "Header Type" field (e.g., bit 7 in this field as illustrated in Chapter 6 of the PCI Local Bus Specification). Other information in the configuration header (e.g., the Device ID register) may indicate to the enumeration process at the I/O processor 14 that one of the device functions is to be concealed. The bus transaction at block 106 may initiate clearing a bit in the Header Type field of the configuration header at the peripheral device 16 to indicate to the host processing

system 12 that the peripheral device 16 has a single device function. At block 108 the host processing system 12 may attempt to configure resources with a first device function associated with the I/O channel as a second device function is being concealed by the modification of the configuration header of the peripheral device 16. With the Header Type field signifying single function device, the host processing terminates the enumeration process for the device after configuring the first function. Nevertheless, this is merely an example of how a processing system may initiate a bus transaction with a peripheral device to conceal a device function of the peripheral device, and embodiments of the present invention are not limited in this respect.

[0029] According to an embodiment, the configuration header of the peripheral device 16 may have write modifiable portions to enable the I/O processor 14 to modify the configuration header to conceal the second function from the host processing system 12. In an embodiment in which the secondary bus 18 is a PCI bus, a bus transaction may set or clear bits indirectly through an extended header register in the configuration header of the peripheral device 16 which is readable and write modifiable from the secondary bus 18. However, this is merely an example of how registers in a configuration header may be write modifiable and embodiments of the present invention are not limited in these respects.

[0030] According to an embodiment, the host processing system 12 may host a device driver to communicate with the peripheral device 16 following an enumeration process. Such a device driver may be configured to communicate with one or more of the device functions of the peripheral device 16 in response to the enumeration process. In some embodiments, the device driver may be configured to communicate with unconcealed device functions even though the device driver is prevented from communicating with the concealed device functions. Here, the host processing system 12 may determine how to configure the device driver based upon modified information in the configuration header data of the peripheral device 16 which may be accessed during an enumeration process. Such modified configuration header data may be a duplication of the aforementioned modified Header Type data in a read-writable Device ID register in an embodiment in which the secondary bus 18 comprises a PCI bus. In an alternative embodiment in which the secondary bus 18 comprises a PCI bus, the I/O processor 14 may modify information in a read-writable extended header register in the configuration header data (e.g., a Device ID register of a configuration header of a PCI

device) of the peripheral device 16. The host processing system 12 may then configure the driver to communicate to the unconcealed device function.

[0031] According to an embodiment, a bridge formed in the I/O processor 14 between the primary bus 24 and secondary bus 18 comprises logic that ensures the I/O processor 14 executes an enumeration procedure to configure resources for devices on the secondary bus 18 prior to completion of an enumeration process at block 108. For example, the bridge may suspend completion of an enumeration procedure to configure processing resources at the host processing system 12 to communicate with devices coupled to the secondary bus 18 until completion of enumeration processes to configure resources at the I/O processor 14 to communicate with devices coupled to the secondary bus 18. In an embodiment in which the primary bus 24 comprises a PCI bus, for example, the bridge may retry Type 1 configuration transactions on the primary bus 24 until the I/O processor 14 completes enumeration of devices on the secondary bus 18. However, this is merely an example of how logic at a bridge may ensure that enumeration of devices and device functions on a secondary bus is completed by a first processing system before completion of enumeration of the devices by a second processing system, and embodiments of the present invention are not limited in these respects.

[0032] Figure 3 shows a schematic diagram of a processing platform 200 comprising a peripheral device 216 according to an alternative embodiment of the present invention. The peripheral device 216 comprises two or more device functions where at least two device functions correspond with the I/O channels 220 and 222. A data bus 224 is coupled to directly to an I/O processor 214, a peripheral device 216 and a host processing system 212 such that the host processing system 212 may enumerate device functions of the peripheral device 216 independently of any bridge coupled between the bus 224 and the peripheral device 216. As in the embodiments discussed above with reference to Figures 1 and 2, the I/O processor 214 may comprise logic to conceal one or more of the device functions of the peripheral device 216 by, for example, modifying read-writable registers in configuration header data at the peripheral device 216 as part of a enumeration procedure (e.g., as discussed above with reference to blocks 102, 104 and 106 in Figure 2). Logic at the host processing system 212 may then configure resources to communicate with the unconcealed device function(s) (e.g., as discussed above with reference to block 108 in Figure 2).

[0033] In the illustrated embodiment, the I/O processor 214 may temporarily inhibit the host processing system 212 from enumerating one or more devices on the data bus 224. The host processing system 212 may enumerate the I/O processor 214 prior to enumerating the peripheral device 216 allowing the I/O processor 214 to enumerate the peripheral device 216. This would enable the I/O processor 214 to initiate one or more bus transactions to enumerate one or more device functions of the peripheral device 216 and to conceal one or more device functions from a subsequent enumeration process initiated at the host processing system 212. In an embodiment in which the data bus 224 comprises a PCI bus, for example, the I/O processor 214 may comprise logic to control the "IDSEL" signal (see, e.g., Sections 3.2.2.3.4 and 3.2.2.3.5 of the PCI Local Bus Specification) on a data interface coupling the data bus 224 and the peripheral device 216. Such logic to control the IDSEL signal may be implemented in, for example, the I/O processor 214 or discrete logic as described in U.S. Patent Appl. Ser. No 09/472,502 filed on December 27, 1999, assigned to Intel Corporation and incorporated herein by reference. For example, the I/O processor 214 may comprise logic to assert an optional PCI signal "TMS" to inhibit the IDSEL signal on the data interface to the peripheral device 216. However, this is merely an example of how a first processing system may temporarily inhibit a second processing system from enumerating a peripheral device on a data bus, and embodiments of the present invention are not limited in this respect.

[0034] While there has been illustrated and described what are presently considered to be example embodiments of the present invention, it will be understood by those skilled in the art that various other modifications may be made, and equivalents may be substituted, without departing from the true scope of the invention. Additionally, many modifications may be made to adapt a particular situation to the teachings of the present invention without departing from the central inventive concept described herein. Therefore, it is intended that the present invention not be limited to the particular embodiments disclosed, but that the invention include all embodiments falling within the scope of the appended claims.